[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 9/48
H04L 9/00

[12]发明专利申请公开说明书

[21]申请号 01103000.3

[43]公开日 2001年8月22日

[11]公开号 CN 1309351A

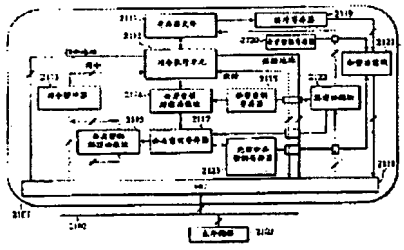
[22]申请日 2001.2.14 [21]申请号 01103000.3
[30]优先权
[32]2000.2.14 [33]JP [31]035898/2000
[32]2000.5.8 [33]JP [31]135010/2000
[71]申请人 株式会社东芝
地址 日本神奈川县
[72]发明人 桥本干生 寺本圭一 齐藤健
白川健治 藤本谦作
[74]专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 吴丽丽

权利要求书4页 说明书43页 附图页数15页

[54]发明名称 抗干预微处理器

[57]摘要

在多任务环境下，抗干预微处理器保存一个其执行被中断的程序 的上下文信息，其中该上下文信息含有指明该程序的执行状态和该程 序的执行码密钥的信息。通过从保存的上下文信息恢复该程序的执行 状态，可以重新启动该程序的执行。利用微处理器的公开密钥可以将 此上下文信息加密，然后利用微处理器的秘密密钥进行解密。



ISSN1008-4274

知识产权出版社出版

权 利 要 求 书

1. 一种具有不能被读出到外部的唯一秘密密钥和与该唯一秘密密钥对应的唯一公开密钥的微处理器，该微处理器包括：

读取单元，被进行配置以从外部存储器读出多个利用不同执行码密钥加密的程序；

解密单元，被进行配置以利用各自解密密钥，对多个通过读取单元读出的程序进行解密；

执行单元，被进行配置以执行多个利用解密单元解密的程序；

上下文信息保存单元，被进行配置以将其执行被中断的一个程序的上下文信息保存到外部存储器或保存到在微处理器内部设置的上下文信息存储器，该上下文信息含有指明此程序的执行状态和此程序的执行码密钥的信息；以及

重新启动单元，被进行配置以通过从外部存储器或上下文信息存储器读出上下文信息并通过从上下文信息中恢复此程序的执行状态，重新启动执行此程序。

2. 根据权利要求 1 所述的微处理器，其中所配置的上下文信息保存单元利用公开密钥对上下文信息进行加密，并将加密上下文信息保存到外部存储器；以及

所配置的重新启动单元通过从外部存储器读出加密上下文信息，利用秘密密钥解密加密上下文信息，以及从解密上下文信息中恢复一个程序的执行状态，重新启动此程序的执行。

3. 根据权利要求 2 所述的微处理器，其中仅当包含在解密上下文信息内的解密执行码密钥与此程序的执行码密钥一致时，重新启动单元才重新启动此程序的执行。

4. 根据权利要求 2 所述的微处理器，其中重新启动单元将包含在解密上下文信息内的解密执行码密钥用作解密密钥以解密此程序。

5. 根据权利要求 1 所述的微处理器，其中所配置的上下文信息保存单元以明文形式将上下文信息保存到此程序被中断后所执行的另一

个程序不可读的上下文信息存储器；以及

通过从上下文信息存储器读出上下文信息并从上下文信息恢复此程序的执行码，所配置的重新启动单元重新启动此程序的执行。

6. 根据权利要求 5 所述的微处理器，其中重新启动单元根据另一个程序规定的指令重新启动此程序的执行。

7. 根据权利要求 6 所述的微处理器，其中在此程序的执行被中断时，上下文信息保存单元将上下文信息保存到上下文信息存储器，并利用公开密钥将上下文信息存储器内的上下文信息加密，然后根据另一个程序规定的另一条指令的执行，将加密上下文信息存储到外部存储器。

8. 根据权利要求 5 所述的微处理器，其中在此程序的执行被中断时，上下文信息保存单元将上下文信息保存到上下文信息存储器，利用公开密钥将上下文信息存储器内的上下文信息加密，然后将加密上下文信息存储到另一个程序规定的外部存储器。

9. 根据权利要求 1 所述的微处理器，其中所配置的上下文信息保存单元产生作为临时密钥的随机数、加密上下文信息、然后将加密上下文信息存储到外部存储器，加密上下文信息含有：第一数值，通过对信息进行加密获得，利用临时密钥指明此程序的执行状态；以及第二数值，通过利用公开密钥加密临时密钥获得；以及

通过从外部存储器读出加密上下文信息，利用秘密密钥由包含在加密上下文信息内的第二数值解密获得临时密钥，利用解密的临时密钥由包含在加密上下文信息内第一数值解密出指明执行状态的信息，以及从解密上下文信息恢复此程序的执行状态，所配置的重新启动单元重新启动此程序的执行。

10. 根据权利要求 9 所述的微处理器，其中上下文信息保存单元保存还含有利用此程序的执行码密钥对临时密钥进行加密获得的第三数值的加密上下文信息。

11. 根据权利要求 10 所述的微处理器，其中重新启动单元利用秘密密钥由包含在加密上下文信息内的第二数值解密获得第一临时密

钥，并利用第一解密临时密钥由包含在加密上下文信息内的第一数值解密获得指明执行状态的信息，同时利用该程序的执行码密钥由包含在加密上下文信息内的第三数值解密获得第二临时密钥，然后只在第一解密的临时密钥与第二解密的临时密钥一致时，重新启动此程序的执行。

12. 根据权利要求 1 所述的微处理器，该微处理器进一步包括：

执行状态存储单元，用于存储当前执行程序的执行状态；以及

执行状态初始化单元，被进行配置以在此程序被中断后而在另一个程序开始之前，将执行状态存储单元的内容初始化为规定数值或将执行状态存储单元的内容加密。

13. 根据权利要求 1 所述的微处理器，该微处理器进一步包括：

密钥读取单元，被进行配置以从外部存储器读出被事先利用公开密钥加密的各程序的执行码密钥；以及

密钥解密单元，被进行配置以利用秘密密钥解密通过密钥读取单元读出的执行码密钥；

其中解密单元利用作为解密密钥的执行码密钥解密各程序。

14. 根据权利要求 1 所述的微处理器，该微处理器进一步包括：

执行状态存储单元，用于存储当前执行程序的执行状态和将被当前执行程序处理的数据的加密属性；以及

数据加密单元，被进行配置以根据存储在执行状态存储单元的加密属性对将由当前执行程序处理的数据进行加密。

15. 根据权利要求 1 所述的微处理器，该微处理器进一步包括：

执行状态存储单元，用于存储当前执行程序的执行状态、将被当前执行程序处理的数据的加密属性以及用于规定加密属性的加密属性规定信息；

相关信息写入单元，被进行配置以将涉及加密属性规定信息并含有利用秘密密钥获得的签名的相关信息写入外部存储器；

相关信息读出单元，被进行配置以根据将由当前执行程序引用的数据的地址从外部存储器读出相关信息；

数据引用许可单元，被进行配置以利用公开密钥验证包含在相关信息内的签名，并且只有当相关信息内的签名与微处理器的原始签名一致时，才允许当前执行程序根据相关信息和规定信息的加密属性，通过确定密钥和用于数据引用的算法进行数据引用；以及

数据加密单元，被进行配置以根据存储在执行状态存储单元的加密属性将由当前执行程序引用的数据加密。

16. 根据权利要求 1 所述的微处理器，该微处理器进一步包括：

高速缓冲存储器，用于以高速缓存行为单位高速缓存多个程序的明文指令和明文数据，该高速缓冲存储器具有属性区用于各高速缓存行，指明在解密其指令被高速缓存到各高速缓存行的各程序或其执行会将明文数据高速缓存到各高速缓存行的各程序时用于唯一标识解密密钥的解密密钥标识符；

高速缓存访问控制单元，被进行配置以只有当加密属性对一个高速缓存行指明的解密密钥标识符与加密属性对另一个高速缓存行指明的解密密钥标识符一致时，允许通过根据另一个高速缓存行内的一个高速缓存数据执行一个存储在一个高速缓存行的高速缓存程序引起的数据引用。

17. 根据权利要求 16 所述的微处理器，其中当不允许进行数据引用时，将新数据从外部存储器高速缓存到另一个高速缓存行。

18. 根据权利要求 16 所述的微处理器，其中当不允许进行数据引用时，保护异常中断此高速缓存程序的执行。

19. 根据权利要求 1 所述的微处理器，其中执行单元还执行明文程序，并具有调试功能块，在明文程序的执行期间，当执行特定地址或地址区域的程序时或将数据引用到特定地址或地址区域的数据时，该调试功能块用于产生异常，在执行加密程序期间，此调试程序无效。

20. 根据权利要求 1 所述的微处理器，其中该微处理器的各组成单元包含在单一芯片或单一封装内。

01.02.14

说　　明　　书

抗干预微处理器

本发明涉及可以在多任务程序执行环境下防止非法变更执行码和非法处理目标数据的微处理器。

最近几年，微处理器的性能得到显著改善，以致微处理器除了具有传统的诸如计算和图形功能外，还可以实现对视频图像和音频声音的再生和编辑。通过在为最终用户设计的系统（以下简称：PC）中实现这种微处理器，用户可以在监视器上欣赏各种视频图像和音频声音。此外，通过将 PC 的再生视频图像和音频声音的功能与计算能力相结合，可以改善对游戏等的适用性。这种微处理器不是专为某种特定硬件设计的而是可以在各种硬件中实现，因此其优势在于，通过简单更换执行程序的微处理器，已经拥有 PC 的用户花费不多就可以欣赏视频图像和音频声音的再生和编辑。

如果在 PC 上处理视频图像和音频声音，就会产生原始图像和音乐的版权保护问题。在 MD 或数字视频重放装置中，通过在这些装置中事先实现防止非法复制的机制，可以防止无限复制。虽然这种装置还在制造，但是试图通过拆除或改变这些装置来进行非法复制的情况却很少，而且世界范围内的趋势是通过法律禁止制造和销售为了进行非法复制能够改变的装置。因此，由于基于硬件进行非法复制造成的损害并不很严重。

然而，在 PC 上对图像数据和音乐数据进行处理是通过软件进行的而不是通过硬件进行的，并且最终用户可以在 PC 上随意改变软件。即，如果用户具有某些知识，则通过分析程序并重写可执行软件，可以非常容易地进行非法复制。此外，不同于硬件的问题是，这样产生的用于非法复制的软件可以通过诸如网络的各种媒体迅速传播。

为了解决这些问题，用于再生诸如商业电影或音乐的版权保护内容的 PC 软件，传统上采用一种通过对软件进行加密防止分析和变更

01.02.14

的技术。这种技术就是抗干预软件（参考 David Aucsmith 等人在 Proceedings of the 1996 Intel Software Developer's Conference 上发表的“Tamper Resistant Software: An Implementation”）。

在防止通过 PC 向用户提供的有价值信息（不仅包括视频数据和音频数据而且包括文本和技术诀窍）的非法复制方面，以及在防止 PC 软件本身的技术诀窍被分析方面，抗干预软件技术仍然有效。

然而，抗干预软件技术是一种，通过在开始执行程序之前对要求保护的程序的一部分进行加密，在执行该部分之前对该部分立即进行解密并在该部分执行完毕后立即对该部分再加密，使得难于利用诸如反汇编程序或调试程序的软件工具进行分析。因此，只要处理器可以执行该程序，通过从程序的启动处开始一步一步进行分析总可以对程序进行分析。

此事实成为版权所有人向系统提供版权保护内容用于利用 PC 再生视频数据和音频数据的障碍。

在这方面，其它抗干预软件应用程序也易受攻击，并且此事实成为通过 PC 进行高级信息服务和将含有企业或个人技术诀窍的程序应用到 PC 的障碍。

总之，在软件保护方面同样存在这些问题，此外，PC 是开放式平台，因此存在通过变更被确定为系统软件配置基础的操作系统（OS）进行攻击问题。换句话说，通过使用属于 OS 的特权，怀有恶意的熟练用户可以变更其自有 PC 的 OS 来废除或分析插入到应用程序内的版权保护机制。

当前的 OS 通过利用对存储器的特权操作功能和 CPU 中提供的特权执行控制功能，在计算机的控制下进行资源管理和资源使用仲裁。管理的目标包括传统目标（例如：设备、CPU 和存储资源）以及网络层或应用层 QoS（服务质量）。尽管如此，资源管理的基础仍然是对执行程序所需的资源进行配置。换句话说，分配 CPU 时间来执行此程序并将分配执行程序所需的存储空间是资源管理的基础。通过控制实现访问这些资源的程序的执行（通过分配 CPU 的时间和存储空间），对

01.02.14

其它设备、网络和应用层服务质量 Qos 进行控制。

OS 具有执行 CPU 时间分配和存储空间分配的特权。换句话说，为了对 CPU 分配时间，OS 具有在任意时间中断并重新启动应用程序的特权并具有在任意时间将分配到应用程序的存储空间的内容转移到不同分层的存储空间的特权。（通常）通过利用应用程序的不同访问速度和访问能力隐匿分层存储系统，将分配到应用程序的存储空间的内容转移到不同分层的存储空间的特权还用于为应用程序提供平面存储器空间。

使用这两种特权，OS 可以中断应用程序的执行状态并在任意时间对它进行快速转储，并且在对它进行拷贝或重写之后重新启动它。此功能还可以被用作分析隐藏在应用程序内的秘密的工具。

为了在计算机上防止应用程序被分析，有几种对程序或数据进行加密的公知技术（例如：Hampson，第 4,847,902 号美国专利、Hartman，第 5,224,166 号美国专利、Davis，第 5,806,706 号美国专利、Takahashi 等，第 5,825,878 号美国专利、Buer 等人，第 6,003,117 号美国专利、第 11－282667 号日本公开专利申请（1999））。然而，这些公知的技术均未涉及防止程序运行过程和数据秘密被 OS 进行上述特权操作问题。

基于 Intel 公司开发的 X86 结构的传统技术（Hartman，第 5,224,166 号美国专利）是一种通过利用规定的密钥 Kx 对执行码和数据进行加密以存储执行码和执行数据的技术。密钥 Kx 可以被表示为 $E_{kp}[Kx]$ 的形式，利用与嵌入到处理器内的秘密密钥 Ks 对应的公开密钥 Kp，可以对 $E_{kp}[Kx]$进行加密。因此，只有知道 Ks 的处理器可以对存储器上的加密执行码进行解密。将密钥 Kx 存储到处理器内被称为段式寄存器的寄存器。

利用这种机制，通过对代码进行加密在某种程度上可以避免用户发现程序代码的秘密。此外，对于不知道代码密钥 Kx 的人来说，由于密码原因难于根据其内涵或利用密钥 Kx 解密时可执行的新产生代码来变更代码。

然而，采用这种技术的系统的缺点在于，利用被称为上下文切换

的 OS 特权有可能对程序进行分析，而无需对加密的执行码进行解密。

更具体地说，当利用中断停止执行程序或当预期系统调用程序自行调用软件中断命令时，为了执行其它程序，OS 进行上下文切换。上下文切换操作将指明该点寄存器值的集合的程序执行状态（以下简称为：上下文信息）存储到存储器，并将事先存储到存储器的另一个程序的上下文信息再存入寄存器。

图 15 示出在 x86 处理器中使用的传统上下文存储格式。这里存储了应用程序使用的寄存器的所有内容。当再启动被中断的程序时，将该程序的上下文信息再存入寄存器。为了并行运行多个程序，上下文切换是不可缺少的功能。在传统技术中，在上下文切换时，OS 可以读取寄存器值，因此根据该程序的执行状态是如何改变的，即使不是全部，也可以猜测出该程序执行的大多数操作。

此外，通过控制在此时通过设置计时器等产生异常的时间，在程序的任意执行点可以进行此处理。除了中断执行和分析之外，还可以恶意重写寄存器信息。重写寄存器不仅可以改变程序运行而且可以使得对程序进行分析更容易。OS 可以存储应用程序的任意状态，因此通过重写寄存器值并通过反复运行程序，可以分析程序的运行。除了上述功能之外，处理器还具有诸如逐步执行的调试支持功能，存在的问题是，利用所有这些功能，OS 可以对应用程序进行分析。

就数据而论，第 5, 224,166 号美国专利认为，仅通过利用加密代码段执行程序，程序可以访问加密数据。这里存在的问题是加密程序利用任意密钥可以自由读取加密数据，而与对程序加密的密钥无关，即使存在利用互相不同的密钥加密的程序。这种传统技术中未说明这些情况，即 OS 和应用程序独立具有它们自己的秘密并且应用程序的秘密不被 OS 发现，或者多个程序供应商分别具有它们自己的秘密。

当然，即使是在现有的处理器中，也可以在应用程序之间划分存储空间并利用虚拟存储机制提供的保护功能来禁止应用程序访问系统存储器。然而，只要虚拟存储制受 OS 的控制，则对应用程序秘密的保护就不能依赖于 OS 控制下的功能。这是由于 OS 可以忽略保护机制

访问数据，并且在提供上述虚拟存储器方面，这种特权不可缺少。

作为另一种传统技术，第 11－282667（1999）号日本公开专利申请公开了一种技术，这种技术为了存储应用程序的秘密信息而在 CPU 内设置秘密存储器。在这种技术中，为了访问秘密存储器内的数据，需要规定基准值。但是，此技术未披露如何防止同一个 CPU 内运行的多个程序（特别是 OS）使用用于获得对秘密数据的访问权的基准数值。

此外，在第 5, 123, 045 号美国专利中，Ostrovsky 等人公开了一种系统，该系统的先决条件是使用具有与应用程序对应的唯一秘密密钥的子处理器，在该系统中，不能根据这些子处理器访问主存储器上的程序的访问方式来推测程序运行。这是基于，通过将根据存储器实现运行的指令系统转换到与此指令系统不同的另一个指令系统，实现随机存储访问的机制。

然而，对不同的应用程序，这种技术要求不同的子处理器，因此这种技术的成本高，并且预期用于处理这种指令系统的编译程序和处理器硬件的执行和快速实现过程非常困难，这是由于它们与当前使用的处理器的编译程序和处理器硬件非常不同。除此之外，与上述说明的将程序码和数据简单加密的其它传统技术（例如：第 5, 224,166 号美国专利和第 11－282667 号日本公开专利申请）比较，在这种处理器中，即使当数据和实际操作码的运行被观察到并被跟踪以致调试程序变得非常困难时，难于包含数据内容与运行之间的对应之处，因此，这种技术存在许多实际问题。

因此，本发明的第一个目的是提供一种微处理器，该微处理器即使是在被中断停止执行时也可以防止在多任务环境下内部执行的算法和存储区内的数据状态被非法分析。

此第一个目的受传统技术能够保护程序码的数值而不能防止利用通过发生异常或调试功能中断程序的执行进行分析的启发。因此，本发明的目的是提供一种即使是在程序执行中断时仍能确实保护代码的微处理器，在此微处理器中，这种保护与当前 OS 要求的执行控制功能和存储器管理功能兼容。

01.02.14

本发明的第二个目的是提供一种即使执行多个利用不同密钥加密的程序，其各程序均可以独立获得正确可读/可写数据区的微处理器。

此第二个目的是受第 5, 224,166 号美国专利公开的传统技术的启发，该技术仅提供简单保护，其中禁止利用非加密代码访问加密数据区，并且不可能独立地对多个程序保护它们的秘密。因此，本发明的目的还在于提供一种当多个应用程序具有它们各自的（加密的）秘密时具有用于防止各应用程序的秘密被 OS 使用的数据区的微处理器。

本发明的第三个目的是提供一种可以防止上述数据区的保护属性（即加密属性）被 OS 非法重写的微处理器。

此第三个目的是受第 5, 224, 166 号美国专利公开的传统技术的启发，该技术的缺点在于，通过利用上下文切换中断程序的执行，OS 可以重写在段式寄存器内设置的加密属性。一旦，通过重写加密属性，程序进入以明文形式写入数据的状态，不加密就不将数据写入存储器。即使在某些时间应用程序校验段式寄存器的数值，但是，如果此后重写寄存器的数值，则结果相同。因此，本发明的目的还在于提供一种微处理器，该微处理器具有可以禁止这种变更或可以检测这种变更并可以对这种变更采取适当措施的机制。

本发明的第四个目的是提供一种微处理器，该微处理器可以防止加密属性受密码分析原理的所谓选择明文攻击法攻击，在密码分析原理中，程序可以使用任意数值作为数据密钥。

本发明的第五个目的是提供一种微处理器，该微处理器具有程序调试和反馈的机制。换句话说，本发明目的在于提供一种处理器，在该微处理器中，在执行失败时，可以以明文的形式调试程序并将关于缺陷的反馈信息送到程序码供应商（程序销售商）。

本发明的第六个目的是提供一种微处理器，该微处理器可以以低成本高性能的形式实现上述第一至第五个目的。

为了实现第一个目的，本发明的第一个方面具有下列特征。通过提供读取功能的总线接口单元，制成单芯片或单封装的微处理器从微处理器外部的存储器（例如：主存储器）读取多个利用代码密钥加密

01.02.14

的程序。对于不同的程序，代码密钥不同。利用分别对应的解密密钥，解密单元对这些读出的程序进行解密，并且指令执行单元执行这些已解密程序。

在中断多个程序中一些程序的执行时，提供执行状态写入功能的上下文信息加密/解密单元利用对微处理器唯一的密钥对指明执行状态的信息进行加密直到中断程序的中断点和代码密钥出现，并将加密的信息作为上下文信息写入微处理器外部的存储器。

如果重新启动被中断的程序，提供重新启动功能的验证单元利用与微处理器的唯一密钥对应的唯一解密密钥解密上下文信息，并只有当包含在已解密上下文信息内的代码密钥（即：预定重新启动程序的代码密钥）与已中断程序的原始代码密钥一致时，才重新执行程序。

此外，为了实现第二和第三个目的，微处理器还具有：存储区（例如：寄存器），它在处理器的内部而且不能被读出到外部；加密属性写入单元（例如：指令 TLB），用于将程序的处理目标数据加密属性写入存储器。加密属性包括程序的代码密钥和加密目标地址范围。在上下文信息中至少含有一部分加密属性。

上下文信息加密/解密单元还将对微处理器唯一的、基于秘密信息的签名附加到上下文信息。这样，验证单元判别解密上下文信息内的签名是否与对微处理器唯一的、基于秘密信息的原始签名一致，如果一致，就重新启动已中断的程序。

同样，将加密程序中断点前的执行状态存储到外部存储器作为上下文信息，而将执行处理目标数据的保护属性存储到处理器内部的寄存器，因此，可以防止非法变更数据。

为了实现第四个目的，本发明的第二个方面具有下列特征。制成单芯片或单封装的微处理器在其内保持不能读出到外部的唯一秘密密